Non-Dictionary-Based Thai Word Segmentation Using Decision Trees

Thanaruk Theeramunkong[1]
Information Technology Program
Sirindhorn International Institute of Technology
Thammasat University, Pathumthani 12121, Thailand
+66-2-986-9103(-8) Ext. 2004
ping@siit.tu.ac.th

Sasiporn Usanavasin
Information Technology Program
Sinrindhorn International Institute of Technology
Thammasat University, Pathumthani 12121, Thailand
+66-2986-9103(-8) Ext. 2002
sasiporn@kind.siit.tu.ac.th

ABSTRACT
For languages without word boundary delimiters, dictionaries are needed for segmenting running texts. This figure makes segmentation accuracy depend significantly on the quality of the dictionary used for analysis. If the dictionary is not sufficiently good, it will lead to a great number of unknown or unrecognized words. These unrecognized words certainly reduce segmentation accuracy. To solve such problem, we propose a method based on decision tree models. Without use of a dictionary, specific information, called syntactic attribute, is applied to identify the structure of Thai words. C4.5 is used as a tool for this purpose. Using a Thai corpus, experiment results show that our method outperforms some well-known dictionary-dependent techniques, maximum and longest matching methods, in case of no dictionary.

Keywords
Decision trees, Word segmentation without a dictionary

1. INTRODUCTION
Word segmentation is a crucial topic in analysis of languages without word boundary markers. Many researchers have been trying to develop and implement in order to gain higher accuracy. Unlike in English, word segmentation in Thai, as well as in many other Asian languages, is more complex because the language does not have any explicit word boundary delimiters, such as a space, to separate between each word. It is even more complicated to precisely segment and identify the word boundary in Thai language because there are several levels and several roles in Thai characters that may lead to ambiguity in segmenting the words. In the past, most researchers had implemented Thai word segmentation systems based on using a dictionary ([2], [3], [4], [6], [7]). When using a dictionary, word segmentation has to cope with an unknown word problem. Up to present, it is clear that most researches on Thai word segmentation with a dictionary suffer from this problem and then introduce some particular process to handle such problem. In our preliminary experiment, we extracted words from a pre-segmented corpus to form a dictionary, randomly deleted some words from the dictionary and used the modified dictionary in segmentation process based two well-known techniques; Maximum and Longest Matching methods. The result is shown in Figure 1. The percentages of accuracy with different percentages of unknown words are explored. We found out that in case of no unknown words, the accuracy is around 97% in both maximum matching and longest matching but the accuracy drops to 54% and 48% respectively, in case that 50% of words are unknown words. As the percentage of unknown words rises, the percentage of accuracy drops continuously. This result reflects seriousness of unknown word problem in word segmentation.

Unknown word (%)	Accuracy (%)	
	Maximum Matching	Longest Matching
0	97.24	97.03
5	95.92	95.63
10	93.12	92.23
15	89.99	87.97
20	86.21	82.60
25	78.40	74.41
30	68.07	64.52
35	69.23	62.21
40	61.53	57.21
45	57.33	54.84
50	54.01	48.67

Figure 1. The accuracy of two dictionary-based systems vs. percentage of unknown words

In this paper, to take care of both known and unknown words, we propose the implementation of a non-dictionary-based system with the knowledge based on the decision tree model ([5]). This model attempts to identify word boundaries of a Thai text. To do

[1] National Electronics and Computer Technology Center (NECTEC), 539/2 Sriyudhya Rd., Rajthevi Bangkok 10400, Thailand

this, the specific information about the structure of Thai words is needed. We called such information in our method as syntactic attributes of Thai words. As the learning stage, a training corpus is utilized to construct a decision tree based on C4.5 algorithm. In the segmentation process, a Thai text is segmented according to the rules produced by the obtained decision tree. The rest shows the proposed method, experimental results, discussion and conclusion.

2. PREVIOUS APPROACHES

2.1 Longest Matching

Most of Thai early works in Thai word segmentation are based on longest matching method ([4]). The method scans an input sentence from left to right, and select the longest match with a dictionary entry at each point. In case that the selected match cannot lead the algorithm to find the rest of the words in the sentence, the algorithm will backtrack to find the next longest one and continue finding the rest and so on. It is obvious that this algorithm will fail to find the correct the segmentation in many cases because of its greedy characteristic. For example:ไปหามเหสี (go to see the queen) will be incorrectly segmented as: ไป(go) หาม (carry) เห(deviate) สี (color), while the correct one that cannot be found by the algorithm is: ไป(go) หา(see) มเหสี (Queen).

2.2 Maximum Matching

The maximum matching algorithm was proposed to solve the problem of the longest matching algorithm describes above ([7]). This algorithm first generates all possible segmentations for a sentence and then select the one that contain the fewest words, which can be done efficiently by using dynamic programming technique. Because the algorithm actually finds real maximum matching instead of using local greedy heuristics to guess, it always outperforms the longest matching method. Nevertheless, when the alternatives have the same number of words, the algorithm cannot determine the best candidate and some other heuristics have to be applied. The heuristic often used is again the greedy one: to prefer the longest matching at each point. For the example, ตาก(expose) ลม(wind) is preferred to ตา(eye) กลม(round).

2.3 Feature-based Approach

A number of feature-based methods have been developed in ([3]) for solving ambiguity in word segmentation. In this approach, the system generates multiple possible segmentation for a string, which has segmentation ambiguity. The problem is that how to select the best segmentation from the set of candidates. At this point, this research applies and compares two learning techniques, called RIPPER and Winnow. RIPPER algorithm is a propositional learning algorithm that constructs a set of rules while Winnow algorithm is a weighted-majority learning algorithm that learns a network, where each node in the network is called a specialist. Each specialist looks at a particular value of an attribute of the target concept, and will vote for a value of the target concept based on its specialty; i.e., based on a value of the attribute it examines. The global algorithm combines the votes from all specialists and makes decision. This approach is a dictionary-based approach. It can acquire up to 91-99% of the number of correct segmented sentences to the total number of sentences.

2.4 Thai Character Chuster

In Thai language, some contiguous characters tend to be an inseparable unit, called Thai character cluster (TCC). Unlike word segmentation that is a very difficult task, segmenting a text into TCCs is easily realized by applying a set of rules. The method to segment a text into TCCs was proposed in ([8]). This method needs no dictionary and can always correctly segment a text at every word boundaries.

3. WORD SEGMENTATION WITH DECISION TREE MODELS

In this paper, we propose a word segmentation method that (1) uses a set of rules to combine contiguous characters to an inseparable unit (syllable-like unit) and (2) then applies a learned decision tree to combine these contiguous units to words. This section briefly shows the concept of TCC and the proposed method based on decision trees.

3.1 Segmenting a Text into TCCs

In Thai language, some contiguous characters tend to be an inseparable unit, called Thai character cluster (TCC). Unlike word segmentation that is a very difficult task, segmenting a text into TCCs is easily recognized by applying a set of rules (in our system, 42 BNF rules). The method to segment a text into TCCs was proposed in [8]. This method needs no dictionary and can always correctly segment a text at every word boundaries. As the first step of our word segmentation approach, a set of rules is applied to group contiguous characters in a text together to form TCCs. The accuracy of this process is 100% in the sense that there is no possibility that these units are divided to two or more units, which are substrings in two or more different words. This process can be implemented without a dictionary, but uses a set of simple linguistic rules based on the types of characters. Figure 2 displays the types of Thai characters. As an example rule, a front vowel and its next consonant must exist in the same unit. Figure 3 shows a fragment of a text segmented into TCCs by the proposed method and its correct word segmentation. Here, a character '|' indicates a segmentation point. The corpus where characters are grouped into TCCs is called a TCC corpus.

Types of Thai Characters	Members
Consonant	ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ฤ ล ฦ ว ศ ษ ส ห ฬ อ ฮ
Upper vowel	◌ิ ◌ี ◌ึ ◌ื ◌ั ◌็ ◌่ ◌้ ◌๊ ◌๋ ◌ำ
Lower vowel	◌ุ ◌ู
Front vowel	เ แ โ ใ ไ
Rear vowel	◌ำ ◌า ◌ๅ ◌ะ ◌ๆ ◌ๅ

Figure 2. Types of Thai characters

TCCs	กา|ร| เก็บ|ภา|มี|ป|ระ|เท|ศ|ไท|ย|และ|ป|ระ|เท|ศ|
CORRECT	การ| เก็บ|ภามี|ประเทศ|ไทย|และ|ประเทศ|

Figure 3. An example of TCCs vs. correct segmentation

3.2 Learning a Decision Tree for Word Segmentation

To learn a decision tree for this task, some attributes are defined for classifying whether two contiguous TCCs are combined to one unit or not. In this paper, eight types of attributes (in Figure 4 are proposed to identify possible word boundaries in the text. The answers (or classes) in the decision tree for this task are of two types: combine and not combine. Moreover, to decide whether two contiguous TCCs should be combined or not, the TCC in front of the current two TCCs and the TCC behind them are taken into account. That is, there are four sets of attributes concerned: two for current two TCCs and two for TCCs in front of and behind the current TCCs. Therefore, the total number of attributes is 32 (that is, 8x4) and there is one dependent variable indicating whether the current two contiguous TCCs should be combined or not.

Attribute Name	Attribute Detail
Front_vowel	0(don't have), 1(don't have rear vowel), 2(may be followed by rear vowel)
Front_consonant	0(don't have), 1(don't lead with hohip or oang), 2(lead with hohip or oang)
Middle_vowel	0(don't have), 1(upper vowel), 2(lower vowel)
Middle_consonant	0(don't have), 1 (have)
Rear_vowel	0(don't have), 1 (sara_a), 2 (sara_aa, sara_am)
Rear_consonant	0-9 are (don't have), (kok_tone), (kod_tone), (kong_tone), (kom_tone), (kob_tone), (kon_tone), (wowaen_tone), (yoyak_tone), (others)
Length	Length of the block (the number of characters)
Space & Enter	0 (don't have), 1 (have)

Figure 4. Types of TCC Attributes

Figure 5 illustrates an example of the process to extract attributes from the TCC corpus and use them as a training corpus. The process is done by investigating the current TCCs in the buffer and recording their attribute values. The dependent variable is set by comparing the combination of the second and the third blocks of characters in the buffer to the same string in the correct word-segmented corpus, the corpus that is segmented by human. The result of this comparison will output whether the second and the third blocks in the buffer should be merged to each other or not. This output is then kept as a training set with the dependent variable, "Combine (1)" or "NotCombine (0)". Repetitively, the start of the buffer is shifted by one block. This process executes until the buffer reaches the end of the corpus. The obtained training set then is used as the input to the C4.5 application ([5]) for learning a decision tree.

The C4.5 program will examine and construct the decision tree using the statistical values calculated from the events occurred. After the decision tree is created, the certainty factor is calculated and assigned to each leaf as a final decision-making factor. This certainty factor is the number that identifies how certain the answer at each terminal node is. It is calculated according to the number of terminal class answers at each leaf of the tree. For example, at leaf node i, if there are ten terminal class answers; six of them are "Combine" and the rest are "Not Combine". The answer at this node would be "Combine" with the certainty factor equals to 0.6 (6/10). On the other hand, leaf node j has 5 elements; two are "Combine" and three are "Not Combine", then the answer at this node would be "Not Combine" with the certainty factor equals to 0.6 (3/5). The general formula for the certainty factor (CF) is shown as follow:

$$CF_i = \frac{\text{Total number of the answer elements at leaf node } i}{\text{Total number of all elements at leaf node } i}$$

We also calculate the recall, precision, and accuracy as defined below:

$$\text{Precision} = \frac{\text{number of correct '|'s in the system answer}}{\text{number of '|'s in the system answer}}$$

$$\text{Recall} = \frac{\text{number of correct '|'s in the system answer}}{\text{number of '|'s in the correct answer}}$$

$$\text{Accuracy} = \frac{\text{number of correct segmented units in system answer}}{\text{total number of segmented units in correct answer}}$$

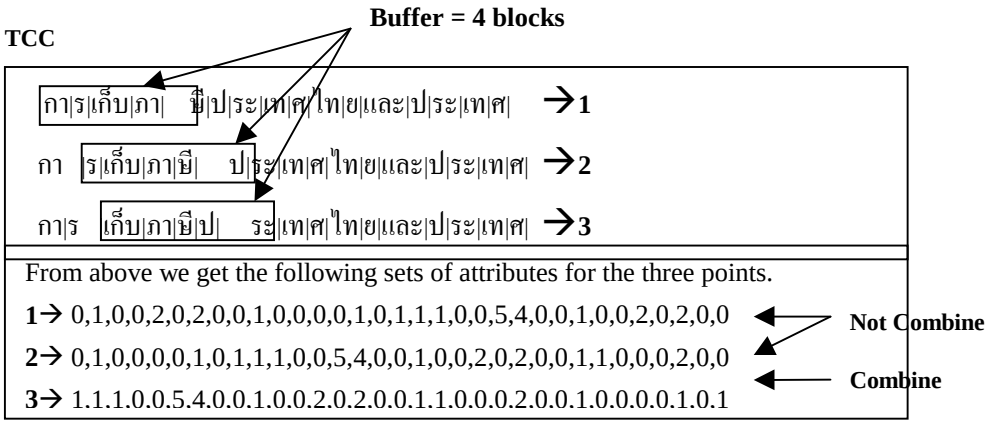

Figure 5. Attributes taken from the corpus

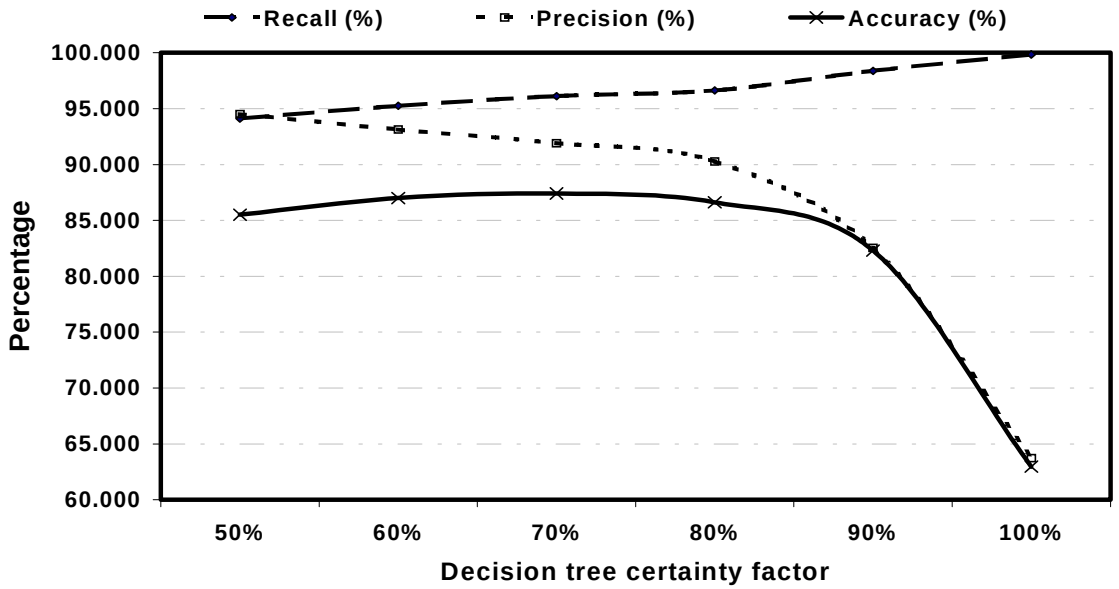

Figure 4. Recall, precision, and accuracy

4. EXPERIMENT RESULTS

In our experiments, the TCC corpus is divided into five sets, four for training and one for testing. Based on this, five times cross validation are performed. To test the accuracy, we trained the decision trees and tested them several times for six different levels of merging permission according to certainty factor(CF). Each level is the starting level of merging permission of the strings in the second and the third blocks in the buffer. Recall, precision, and accuracy where the certainty factor ranges between 50% and 100% are shown in Figure 6.

From the result, we observed that our method presented the satisfactory in the percentage of accuracy and both in precision and recall compared to those numbers of the original TCC performance. The TCC corpus has 100% recall but has 52.12% precision, and 44.93% accuracy. Using the decision tree learned from a Thai corpus, the precision improves up to 94.11-99.85% and the accuracy increases up to 85.51-87.41%. However, the recall drops to 63.72-94.52%. For a high CF, say 100 %, recall drops a little because there are few cases to merge two TCCs but precision and accuracy improve dominantly to 63.72% and 62.97, respectively. For a lower CF, say 50%, recall drops dominantly but precision and accuracy dramatically improve to 94.52% and 85.51% respectively.

However, from 50 to 100% CF, at approximately 80% CF, the accuracy had declined. The reason to this declination is that with a very high level of merging permission, there are a few chances for removing '|' because of the %CF at those leaves are lower than this permission level. Therefore, there are more chances for wrong word segmentation, which lead to decrease accuracy. In conclusion, the appropriate level of merging permission has to be used in order to achieve high accuracy. From our experiment, the best permission level is approximately equal to 70%, which gives the recall equals to 96.13%, precision equals to 91.92% and the accuracy equals to 87.41%.

5. DISCUSSION AND CONCLUSION

Due to the problem of the unknown words that most of the existing Thai word segmentation systems have to cope with, this paper has introduced an alternative method for avoiding such problem. Our approach is based on using the decision tree as the decision support model with no need of dictionary at all. The experimental results clearly show that our method gives some promises on achieving high accuracy when suitable and appropriate merging permission factor is used. In our experiments, the best level of permission that leads to the highest accuracy is approximately equals to 70%, which gives the accuracy equal to 87.41%, as shown in Figure 6.

The dictionary-based method so-called the feature-based system with context independence gives the highest accuracy equals to 99.74% and with context dependence, which has the highest accuracy equals to 95.33% ([3]). In [1], the Japanese word segmentation is explored based on decision tree. However, it focuses on the part-of-speech for word segmentation. Another two well known dictionary-based methods, Maximum and Longest Matching methods, have the accuracy equal to 86.21% and 82.60% respectively when there are 20% of unknown words, which are lower than our system accuracy, and their accuracy drops as percentage of unknown words increases. By comparing these percentages of accuracy, we can conclude that our method can achieve satisfied accuracy even without dictionary. Therefore, our method is useful for solving an unknown word problem and it will be even more useful to apply our method to the dictionary-based system in order to improve the system accuracy. In addition, our results seem to suggest that our method is efficient not only for Thai texts but also for any language when suitable and appropriate syntactic attributes are used.

Our plan for further research is to apply our method to the dictionary based system in order to take care of the unknown word parts. This would improve the accuracy of the system regardless of the level of the unknown words found in the context.

6. ACKNOWLEDGEMENT

This work has been supported by National Electronics and Computer Technology Center (NECTEC) under the project number NT-B-06-4F-13-311.

7. REFERENCES

[1] Kasioka, H., Eubank, S. G., and Black, E. W., Decision-Tree Morphological Analysis without a Dictionary for Japanese, Proceedings of the Natural Language Processing Pacific Rim Symposium, pp. 541-544, Phuket, Thailand, 1997.

[2] Kawtrakul, A., Thumkanon, C., Poovorawan, Y., Varasrai, P. and Suktarachan, M., Automatic Thai Unknown Word Recognition, Proceedings of the Natural Language Processing Pacific Rim Symposium, pp. 341-348, Phuket, Thailand, 1997.

[3] Mekanavin, S., Charenpornsawat, P., and Kijsirikul, B., Feature-based Thai Words Segmentation, Proceedings of the Natural Language Processing Pacific Rim Symposium, pp. 41-48, Phuket, Thailand, 1997.

[4] Poowarawan, Y., Dictionary-based Thai Syllable Separation, Proceedings of the Ninth Electronics Engineering Conference, 1986.

[5] Quinlan, J.R., Induction of Decision Trees, Machine Learning, 1, pp. 81-106, 1986.

[6] Rarunrom, S. Dictionary-based Thai Word Separation, Thesis, Thailand.

[7] Sornlertlamvanich, V., Word Segmentation for Thai in a Machine Translation system (in Thai), Papers on Natural Language processing, NECTEC, Thailand, 1995.

[8] Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., Chinnan, W., Character-Cluster Based Thai Information Retrieval, Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages, September 30 - October 20, 2000, Hong Kong, pp.75-80.